REMARKS

This application has been carefully considered in connection with the Examiner's Office Action dated June 23, 2008. Reconsideration and allowance are respectfully requested in view of the following.

Summary of Rejections

Claims 1 – 26 were pending at the time of the Office Action

Claims 1 – 8, 10, 11, 17 – 20, and 24 – 26 were rejected under 35 U.S.C. 103(a) as being unpatentable over US 7,007,278 (hereinafter "Gungabeesoon") in view of US 6,757,746 (hereinafter "Boucher").

Claim 9 was rejected under 35 U.S.C. 103(a) as being unpatentable over Gungabeesoon in view of Boucher as applied to Claim 7 and further in view of US 5,721,876 (hereinafter "Yu").

Claims 12-14 were rejected under 35 USC § 103(a) as being unpatentable over Gungabeesoon in view of Boucher as applied to claim 6 above, and further in view of U.S. Patent No. 6,931,623 ("Vermeire").

Claims 15 and 16 were rejected under 35 U.S.C. 103(a) as being unpatentable over Gungabeesoon in view of Boucher as applied to Claim 6 and further in view of US 5,745,748 (hereinafter "Ahmad").

Claims 21 – 23 were rejected under 35 U.S.C. 103(a) as being unpatentable over Gungabeesoon in view of Yu and Boucher.

10

Summary of Response

Claims 1, 6, 11, 19, and 21 – 23 were previously presented.

Claims 3, 8, and 24 are currently amended.

Claims 27, 28, and 29 are new.

Claims 4, 5, and 9 are canceled herein.

Claims 2, 7, 10, 12 – 18, 20, 25, and 26 remain as originally submitted.

Remarks and Arguments are provided below.

Summary of Claims Pending

Claims 1-3, 6-8, and 10-29 are currently pending following this response.

Applicant Initiated Interview

Applicant thanks Examiner Chen and Primary Examiner Zhen for their time and consideration of the proposed amendments presented in the interview on September 9, 2008. In the interview, Applicant discussed possible interpretations of the limitation of a bit-level call and further discussed proposed claim amendments to further clarify the definition of a bit-level call to the operating system. Applicant also reiterated its request for clarification on how the disclosures of the Gungabeesoon and Boucher references were combined in the Office Action. Examiner Chen and Primary Examiner Zhen indicated that further search and/or consideration may be required upon receiving this response.

11

Response to Rejections

Gungabeesoon in view of Boucher does not teach or suggest a routine callable from a COBOL program that reads information from a socket or pipe through a bit-level call to an operating system. By reading information from the socket or pipe, the routine of the pending application enables the COBOL program to have access to information on the socket or pipe, to which the COBOL program would not natively have access. Therefore, the routine enables COBOL programs to communicate in a distributed environment.

COBOL does not natively support distributed functions such as sockets and pipes. The pending application discloses a COBOL application using a technical layer to make a bit-level call to an operating system. For example, a COBOL routine in a technical layer may read information from a socket through a bit level call to the operating system, wherein the bit-level call includes correct bits, offsets, and memory mapping to sufficiently interface with the operating system. Therefore, the technical layer, which may or may not be written in COBOL, enables the COBOL application to gain access to information on sockets and pipes and thereby communicate in a distributed environment.

Gungabeesoon describes a legacy application that executes normally and is unaware of any changes to its native environment. Gungabeesoon discloses that the I/O of the legacy application is intercepted and redirected by the operating system such that the legacy application may be used over a network without requiring code changes to the legacy application. Gungabeesoon does not disclose a routine callable by the legacy

12

application that reads information from a socket or pipe through a bit-level call to the operating system.

Boucher describes an application named Samba, which is not a COBOL program or other legacy application. Rather, the Samba application of Boucher is a more modern application natively enabled to run in distributed, networked environments. Boucher describes Samba initializing an application-to-operating system communication session by calling a socket function. The socket-calling function in Boucher is not a COBOL routine. Further, while Boucher may disclose that the socket function causes the kernel of the operating system to perform a function, the socket function is not a bit-level call. Rather, the socket function of the Samba application appears to be a system call.

These distinctions, as well as others, will be discussed in greater detail in the analysis of the present claims that follows.

Response to Rejections under Section 103

Claim 1:

In the Office Action dated June 23, 2008, claim 1 was rejected under 35 USC § 103(a) as being unpatentable over Gungabeesoon in view of Boucher.

I.   Gungabeesoon in view of Boucher does not teach or suggest a COBOL routine callable from a COBOL program that reads information from a socket.

Claim 1 recites, "a COBOL routine callable from the COBOL program, the COBOL routine reads information from the socket and writes the information read from the socket

13

to the memory block in response to the COBOL program call, wherein the COBOL routine reads the information from the socket through a bit-level call to an operating system".

The Office Action concedes on page 3 that Gungabeesoon does not provide the limitations of claim 1 cited above. The Office Action relied on the disclosures in Column 3, line 55 – Column 4, line 4 of Boucher to supplement the shortcomings of Gungabeesoon. The cited section of Boucher is reproduced below:

> FIG. 3 is a flowchart of a method in accordance with one specific embodiment of the present invention. In a first step (step 300), the Samba application program 104 initializes application-to-operating system communication by calling the "socket" function. The socket function causes kernel 105 to allocate a communication control block (CCB) that will be used to manage the connection. The Samba application program 104 then uses the "bind" routine to associate the socket with a particular local IP address and IP port. The Samba application program 104 then calls the "listen" routine to wait for an incoming connection to arrive from kernel 105. When an incoming connection arrives, the Samba application program 104 calls the "accept" routine to complete the connection setup. After setting up the socket, the Samba application 104 uses the "select" routine to tell the kernel 105 to alert application 104 when data for that particular connection has arrived.

As shown above, Boucher describes an application named Samba involved in the transmission of information between hosts. But the Samba application is a modern application that is natively network-aware. Samba is an open source implementation of networking protocols to share files between computers. Samba is natively able to call socket functions without any outside software components providing support. In contrast, claim 1 requires, "a COBOL routine callable from the COBOL program." The Samba application is not a COBOL program and the socket function used by the Samba

14

application is not a COBOL routine. In particular, as a natively network-aware application, Samba does contain socket functionality. But Samba's socket functionality is not a COBOL routine. Boucher only discloses in passing that an application may call a socket function which may cause an operating system to perform some function. Boucher contains no reference to the COBOL programming.

Accordingly, Applicants respectfully submit that Gungabeesoon in view of Boucher does not teach or suggest a COBOL routine callable by a COBOL application that reads information from a socket and thereby gains distributed networking functionality for the COBOL application.

II.   <u>Gungabeesoon in view of Boucher does not teach or suggest a COBOL routine that reads information from a socket through a bit-level call to an operating system.</u>

Claim 1 recites, "... wherein the COBOL routine reads the information from the socket through a bit-level call to an operating system".

The Office Action relies on the socket function of Boucher to read on the bit-level call to the operating system in claim 1 of the present disclosure. The Office Action states on page 4, "In other words, the socket function has to make "bit-level" calls to the operating system in order to interface with the operating system according to its kernel system call requirements." While the socket function as described by Boucher is called to initialize application-to-operating system communication and "causes kernel 105 to allocate a communication control block (CCB) that will be used to manage the connection", this is not a bit-level call. Rather, as noted by the Office Action, the socket

15

function disclosed by Boucher is a system call. One skilled in the art at the time of the invention would recognize that system calls are high-level or assembly language calls.

For further clarity on the claim limitation of a "bit-level call" with reference to the pending specification, Applicant again refers to the "Clarification of Claim Limitations" section on pages 10 – 12 of the Response filed on March 26, 2008. Further, for clarity of the ordinary meanings of the words in the claim limitation of a "bit-level call", Applicant notes the definitions provide by the Merriam-Webster Online Dictionary.

Regarding the word "bit", the etymological root of the word "bit' is the phrase "binary digit." The Merriam-Webster Online Dictionary defines "bit" as "a unit of computer information equivalent to the result of a choice between two alternatives (as yes or no, on or off)." One skilled in the art at the time of the invention would recognize that a bit is typically represented in computer code as a "1" or a "0".

The Merriam-Webster Online Dictionary defines the word "level" as "a position in a scale or rank." One skilled in the art at the time of the invention would recognize that there are many levels of computer program code. For example, computer program code may be programmed by a user in a high-level language (e.g. C, C++, etc.). The high-level programming code may then be compiled by a compiler into machine code which is bit patterns or bit sequences that may be executed by a CPU.

According to the ordinary meaning of the terms, the limitation of "bit-level" should be interpreted as on a position in scale or rank with bits (e.g. computer code representations of a binary digit such as a "1" or "0"). Therefore, according to the ordinary

16

meanings of the terms, a "bit-level call" should be interpreted as a call made on a position in scale or rank with bits. In other words, a call including a sequence of 1's and 0's. Applicant notes that this is consistent with the specification in Paragraph 041. Applicant further notes that a high-level or assembly language call is not a call including a sequence of 1's and 0's.

Accordingly, the socket function disclosed by Boucher cannot be interpreted as a bit-level call to an operating system, as claimed.

For at least the reasons established above in sections I and II above, Applicant respectfully submits that independent claim 1 is not taught or suggested by Gungabeesoon in view of Boucher and respectfully requests allowance of this claim.

Claims Depending From Claim 1:

Claims 2-5 were rejected under 35 USC §103(a) as being unpatentable over Gungabeesoon in view of Boucher.

Claim 27 has been added by this amendment. Applicant respectfully submits that no new matter has been added. Support may be found throughout the specification as originally filed, including at least paragraph 041.

Dependent claims 4 and 5 have been canceled herein. Dependent claims 2, 3, and 27 depend directly or indirectly from independent claim 1 and incorporate all of the limitations thereof. Accordingly, for at least the reasons established in sections I and II above, Applicant respectfully submits that claims 2, 3, and 27 are not taught or suggested

17

by Gungabeesoon in view of Boucher and respectfully requests allowance of these claims.

Claim 27:

III.   Gungabeesoon in view of Boucher does not teach or suggest making a bit-level call to an operating system wherein the bit-level call includes the correct bits, offsets, and memory mapping to sufficiently interface with the operating system.

New dependent claim 27 has been added by this amendment to further clarify that bit-level calls to an operating system have the correct bits, offsets, and memory mapping to sufficiently interface with the operating system. This amendment is supported by paragraph 041 of the specification which states that "operating system calls require bit level mapping of the calls, parameters and returned information to complete a COBOL programming language call." Paragraph 041 also describes the use of bit level calls "to communicate with the operating system to enable the COBOL program to look like an assembler call, as necessitated by the operating system." Paragraph 041 further discloses, "As such, the call to the operating system 34 has the correct bits, offsets and memory mapping to sufficiently interface with the operating system 34."

As discussed above, Gungabeesoon does not disclose a bit-level call to an operating system and Boucher teaches an application initializing application-to-operating system communication by calling a socket function. The socket function of Boucher causes the kernel of the operating system to allocate a communication control block that will be used to manage the connection.

18

The socket function of Boucher is not a bit-level call. Further, Applicant respectfully submits that Gungabeesoon in view of Boucher does not teach or suggest that a bit-level call includes the correct bits, offsets and memory mapping to sufficiently interface with the operating system. Applicant respectfully submits that the Samba application natively incorporating a socket-calling functionality is fundamentally different from a COBOL program calling a COBOL routine which in turn makes a bit-level call including the correct bits, offsets and memory mapping to sufficiently interface with the operating system to access socket functionality. Accordingly, Gungabeesoon in view of Boucher does not teach or suggest making a bit-level call to an operating system, wherein the bit-level call includes the correct bits, offsets, and memory mapping to sufficiently interface with the operating system.

In addition to the reasons established above, for at least the reasons established in section III above, Applicant respectfully submits that dependent claim 27 is not taught or suggested by Gungabeesoon in view of Boucher and respectfully requests allowance of this claim.

Claim 6:

Claim 6 was rejected under 35 USC §103(a) as being unpatentable over Gungabeesoon in view of Boucher.

Claim 6 includes limitations substantially similar to the limitations discussed in sections I and II above. For example, claim 6 recites, "… retrieving, by a COBOL routine

19

..., information from the socket through a bit-level call to an operating system." For at least the reasons established above in sections I and II above, Applicant respectfully submits that independent claim 6 is not taught or suggested by Gungabeesoon in view of Boucher and respectfully requests allowance of this claim.

Claims Depending From Claim 6:

Claims 7, 8, 10, 11, and 17 – 20 were rejected under 35 U.S.C. 103(a) as being unpatentable over Gungabeesoon in view of Boucher.

Claim 9 was rejected under 35 U.S.C. 103(a) as being unpatentable over Gungabeesoon in view of Boucher as applied to Claim 7 and further in view of Yu.

Claims 12-14 were rejected under 35 USC § 103(a) as being unpatentable over Gungabeesoon in view of Boucher as applied to claim 6 above, and further in view of Vermeire.

Claims 15 and 16 were rejected under 35 U.S.C. 103(a) as being unpatentable over Gungabeesoon in view of Boucher as applied to claim 6 and further in view of Ahmad.

Claim 28 has been added herein. Applicant respectfully submits that no new matter has been added. Support may be found throughout the specification as originally filed, including at least paragraph 041. Claim 9 has been canceled herein.

Dependent claim 9 has been canceled herein. Dependent claims 7, 8, 10 – 20, and 28 depend directly or indirectly from independent claim 6 and incorporate all of the

limitations thereof. Accordingly, for at least the reasons established in sections I and II above, Applicant respectfully submits that dependent claims 7, 8, 10 – 20, and 28 are not taught or suggested by Gungabeesoon in view of Boucher and respectfully request allowance of these claims. Neither Yu, Vermeire nor Ahmad, alone or in combination, cure the deficiencies of Gungabeesoon in view of Boucher.

Claim 28:

Claim 28 includes limitations substantially similar to the limitations discussed in section III above. In addition to the reasons established above, for at least the reasons established in section III, Applicant respectfully submits that dependent claim 28 is not taught or suggested by Gungabeesoon in view of Yu and Boucher and respectfully requests allowance of this claim. Applicant respectfully submits that Yu does not cure the deficiencies of Gungabeesoon and Boucher.

Claim 21:

Claim 21 was rejected under 35 USC §103(a) as being unpatentable over Gungabeesoon in view of Yu and Boucher.

Claim 21 includes limitations substantially similar to the limitations discussed in sections I and II above. For example, claim 21 recites, "wherein the COBOL routine reads the information from the pipe through a bit-level call to an operating system." For at least the reasons established in sections I and II above, Applicant respectfully submits

21

that independent claim 21 is not taught or suggested by Gungabeesoon in view of Yu and Boucher and respectfully requests allowance of this claim.

Claims Depending From Claim 21:

Claims 22 and 23 were rejected under 35 USC §103(a) as being unpatentable over Gungabeesoon in view of Yu and Boucher.

Dependent claim 29 has been added herein. Applicant respectfully submits that no new matter has been added. Support may be found throughout the specification as originally filed, including at least paragraph 041.

Dependent claims 22, 23, and 29 depend directly or indirectly from independent claim 21 and incorporate all of the limitations thereof. Accordingly, for at least the reasons established in sections I and II above, Applicant respectfully submits that dependent claims 22, 23, and 29 are not taught or suggested by Gungabeesoon in view of Yu and Boucher and respectfully request allowance of these claims.

Claim 29:

Claim 29 includes limitations substantially similar to the limitations discussed in section III above. In addition to the reasons established above, for at least the reasons established in section III, Applicant respectfully submits that dependent claim 29 is not taught or suggested by Gungabeesoon in view of Yu and Boucher and respectfully requests allowance of this claim. Applicant respectfully submits that Yu does not cure the deficiencies of Gungabeesoon and Boucher.

22

Claim 24:

Claim 24 was rejected under 35 USC §103(a) as being unpatentable over Gungabeesoon in view of Boucher.

Claim 24 includes limitations substantially similar to the limitations discussed in sections II and III above.   For example, claim 24 provides "a method for socket communication in COBOL comprising reading ... information from a socket through a bit-level call to an operating system, wherein the bit-level call to the operating system includes correct bits, offsets, and memory mapping to sufficiently interface with the operating system." For at least the reasons established above in sections II and III above, Applicant respectfully submits that independent claim 24 is not taught or suggested by Gungabeesoon in view of Boucher and respectfully requests allowance of this claim.

Claims Depending From Claim 24:

Claims 25 and 26 were rejected under 35 USC §103(a) as being unpatentable over Gungabeesoon in view of Boucher.

Dependent claims 25 and 26 depend directly or indirectly from independent claim 24 and incorporate all of the limitations thereof. Accordingly, for at least the reasons established in sections II and III above, Applicant respectfully submits that dependent claims 25 and 26 are not taught or suggested by Gungabeesoon in view of Boucher and respectfully request allowance of these claims.

23

CONCLUSION

Applicant respectfully submits that the present application is in condition for allowance for the reasons stated above. If the Examiner has any questions or comments or otherwise feels it would be helpful in expediting the application, he is encourage to telephone the undersigned at (972) 731-2288.

The Commissioner is hereby authorized to charge payment of any further fees associated with any of the foregoing papers submitted herewith, or to credit any overpayment thereof, to Deposit Account No. 21-0765, Sprint.

Respectfully submitted,

Date: September 22, 2008

CONLEY ROSE, P.C.
5601 Granite Parkway, Suite 750
Plano, Texas 75024
(972) 731-2288
(972) 731-2289 (facsimile)

/Michael W. Piper/
Michael W. Piper
Reg. No. 39,800

ATTORNEY FOR APPLICANT

24